

Parsiranje od vrha ka dnu

Zadatak 1. (Primer sa predavanja)

Jedan prepoznavać od vrha ka dnu, na osnovu sledeće gramatike, prepoznaje smene za jednu ulaznu sekvencu po sledećem redosledu: 1, 4, 3, 2, 4, 2, 1, 4, 2, 4, 4.

O kojoj ulaznoj sekvenci je reč?

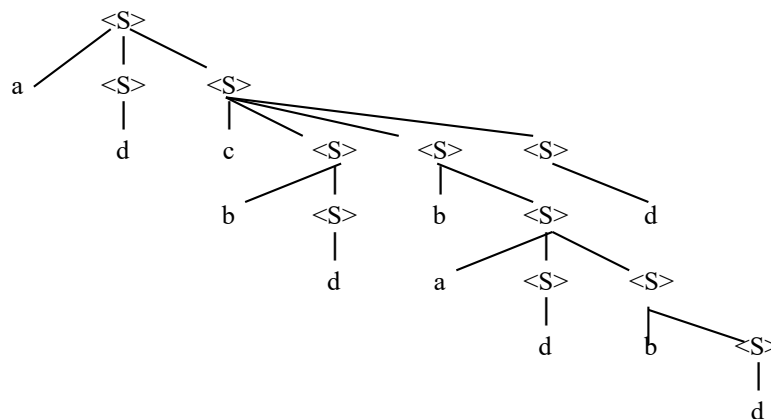
- | | |
|--|--|
| 1. $\langle S \rangle \rightarrow a \langle S \rangle \langle S \rangle$ | 3. $\langle S \rangle \rightarrow c \langle S \rangle \langle S \rangle \langle S \rangle$ |
| 2. $\langle S \rangle \rightarrow b \langle S \rangle$ | 4. $\langle S \rangle \rightarrow d$ |

Rešenje

Prepoznavać od vrha ka dnu prepoznaje smene u redosledu koji odgovara levom izvođenju tražene ulazne sekvence počev od startnog neterminala.

$$\begin{aligned}
 &\langle S \rangle \Rightarrow_{lm} a \langle S \rangle \langle S \rangle \Rightarrow_{lm} ad \langle S \rangle \Rightarrow_{lm} adc \langle S \rangle \langle S \rangle \langle S \rangle \Rightarrow_{lm} adcb \langle S \rangle \langle S \rangle \langle S \rangle \Rightarrow_{lm} \\
 &\quad \uparrow \quad \quad \uparrow \quad \quad \quad \uparrow \quad \quad \quad \uparrow \quad \quad \quad \uparrow \\
 &\quad 1 \quad \quad 4 \quad \quad \quad 3 \quad \quad \quad 2 \quad \quad \quad 4 \\
 &\Rightarrow_{lm} adcbdb \langle S \rangle \langle S \rangle \Rightarrow_{lm} adcbdb \langle S \rangle \langle S \rangle \Rightarrow_{lm} adcbdba \langle S \rangle \langle S \rangle \langle S \rangle \Rightarrow_{lm} \\
 &\quad \quad \quad \uparrow \quad \quad \quad \uparrow \quad \quad \quad \uparrow \\
 &\quad \quad 2 \quad \quad \quad 1 \quad \quad \quad 4 \\
 &\Rightarrow_{lm} adcbdbad \langle S \rangle \langle S \rangle \Rightarrow_{lm} adcbdbadb \langle S \rangle \langle S \rangle \Rightarrow_{lm} \\
 &\quad \quad \quad \uparrow \quad \quad \quad \uparrow \\
 &\quad \quad 2 \quad \quad \quad 4 \\
 &\Rightarrow_{lm} adcbdbadbdb \langle S \rangle \Rightarrow_{lm} adcbdbadbdbd \\
 &\quad \quad \quad \uparrow \\
 &\quad \quad 4
 \end{aligned}$$

Traženi ulazni niz je adcbdbadbdbd. Odgovarajuće stablo izvođenja je:



Zadatak 2.

Za sledeću gramatiku sa startnim simbolom $\langle S \rangle$, pronaći FOLLOW skupove svih neterminala; pronaći SELECT skupove svih produkcija i odgovoriti na pitanje da li se radi o LL(1) gramatici.

1. $\langle S \rangle \rightarrow a \langle A \rangle \langle B \rangle b \langle C \rangle \langle D \rangle$
2. $\langle S \rangle \rightarrow \varepsilon$
3. $\langle A \rangle \rightarrow \langle A \rangle \langle S \rangle d$
4. $\langle A \rangle \rightarrow \varepsilon$
5. $\langle B \rangle \rightarrow \langle S \rangle \langle A \rangle c$
6. $\langle B \rangle \rightarrow e \langle C \rangle$
7. $\langle B \rangle \rightarrow \varepsilon$
8. $\langle C \rangle \rightarrow \langle S \rangle f$
9. $\langle C \rangle \rightarrow \langle C \rangle g$
10. $\langle C \rangle \rightarrow \varepsilon$
11. $\langle D \rangle \rightarrow a \langle B \rangle \langle D \rangle$
12. $\langle D \rangle \rightarrow \varepsilon$

Analiza problema

Bezkontekstna gramatika pripada klasi LL(1) ako i samo ako smene te gramatike sa istom levom stranom imaju disjunktne selekzione skupove.

Selekциони skupovi za LL(1) gramatike definišu se na sledeći način:

Za proizvoljnu produkciju $\langle A \rangle \rightarrow \alpha$, važi da je

$\text{SELECT}(\langle A \rangle \rightarrow \alpha) = \text{FIRST}(\alpha)$, ako α nije poništiva sekvenca, ili

$\text{SELECT}(\langle A \rangle \rightarrow \alpha) = \text{FIRST}(\alpha) \cup \text{FOLLOW}(\langle A \rangle)$, u suprotnom.

Koraci u rešavanju ovog zadatka biće:

1. Pronalaženje poništivih neterminala.
2. Određivanje FIRST skupova za svaki neterminal.
3. Određivanje FOLLOW skupova za sve neterminale.
4. Određivanje selekcionih skupova smena.

Rešenje

1. Pronalaženje poništivih neterminala.

Poništivi neterminali: $\langle S \rangle$, $\langle A \rangle$, $\langle B \rangle$, $\langle C \rangle$, $\langle D \rangle$

2. Određivanje FIRST skupova za svaki neterminal.

$\text{FIRST}(\langle S \rangle) = \{a\}$

$\text{FIRST}(\langle A \rangle) = \text{FIRST}(\langle A \rangle \langle S \rangle d) = \text{FIRST}(\langle A \rangle) \cup \text{FIRST}(\langle S \rangle) \cup \{d\} = \{a, d\}$

$\text{FIRST}(\langle B \rangle) = \text{FIRST}(\langle S \rangle) \cup \text{FIRST}(\langle A \rangle) \cup \{c\} \cup \{e\} = \{a, c, d, e\}$

$\text{FIRST}(\langle C \rangle) = \text{FIRST}(\langle S \rangle) \cup \{f\} \cup \text{FIRST}(\langle C \rangle) \cup \{g\} = \{a, f, g\}$

$\text{FIRST}(\langle D \rangle) = \{a\}$

3. Određivanje FOLLOW skupova za sve neterminale.

$\text{FOLLOW}(\langle S \rangle) \supset \{ \mid \}$

$\supset \{d\}$

$\supset \text{FIRST}(\langle A \rangle c) = \text{FIRST}(\langle A \rangle) \cup \{c\}$

$\supset \{f\}$

$\text{FOLLOW}(\langle A \rangle) \supset \text{FIRST}(\langle B \rangle) \cup \{b\}$

$\supset \text{FIRST}(\langle S \rangle) \cup \{d\}$

$\supset \{c\}$

$\text{FOLLOW}(\langle B \rangle) \supset \{b\}$

$\supset \text{FIRST}(\langle D \rangle) \cup \text{FOLLOW}(\langle D \rangle)$

$\text{FOLLOW}(\langle C \rangle) \supset \text{FIRST}(\langle D \rangle) \cup \text{FOLLOW}(\langle S \rangle)$

$$\supset \text{FOLLOW}(\langle B \rangle)$$

$$\supset \{g\}$$

$$\text{FOLLOW}(\langle D \rangle) \supset \text{FOLLOW}(\langle S \rangle) \cup \text{FOLLOW}(\langle D \rangle)$$

Na osnovu prethodnih nejednakosti se dobija:

$$\text{FOLLOW}(\langle S \rangle) = \{a, c, d, f, \mid\}$$

$$\text{FOLLOW}(\langle A \rangle) = \{a, b, c, d, e\}$$

$$\text{FOLLOW}(\langle B \rangle) = \{a, b, c, d, f, \mid\}$$

$$\text{FOLLOW}(\langle C \rangle) = \{a, b, c, d, f, g, \mid\}$$

$$\text{FOLLOW}(\langle D \rangle) = \{a, c, d, f, \mid\}$$

4. Određivanje selekcionih skupova smena na način opisan u analizi problema.

$$\text{SELECT}(1) = \{a\}$$

$$\text{SELECT}(2) = \{a, c, d, f, \mid\}$$

$$\text{SELECT}(3) = \{a, d\}$$

$$\text{SELECT}(4) = \{a, b, c, d, e\}$$

$$\text{SELECT}(5) = \{a, d, c\}$$

$$\text{SELECT}(6) = \{e\}$$

$$\text{SELECT}(7) = \{a, b, c, d, f, \mid\}$$

$$\text{SELECT}(8) = \{a, f\}$$

$$\text{SELECT}(9) = \{a, f, g\}$$

$$\text{SELECT}(10) = \{a, b, c, d, f, g, \mid\}$$

$$\text{SELECT}(11) = \{a\}$$

$$\text{SELECT}(12) = \{a, c, d, f, \mid\}$$

Gramatika nije LL(1), jer, na primer, selekциони skupovi 1. i 2. smene nisu disjunktni.

Zadatak 3.

a) Eliminirati mrtve i nedostižne neterminale iz date gramatike sa startnim simbolom $\langle A \rangle$.

b) Iz gramatike dobijene pod a) eliminirati levu rekurziju.

c) Izvršiti levu faktorizaciju u gramatici dobijenoj pod b).

c) Izračunati selekzione skupove smena gramatike dobijene pod c). Da li se radi o LL(1) gramatici?

$$1. \langle A \rangle \rightarrow \langle A \rangle c \langle B \rangle$$

$$7. \langle C \rangle \rightarrow \langle C \rangle c \langle E \rangle$$

$$2. \langle A \rangle \rightarrow \langle C \rangle c b$$

$$8. \langle D \rangle \rightarrow \langle D \rangle a \langle B \rangle$$

$$3. \langle A \rangle \rightarrow c \langle D \rangle$$

$$9. \langle D \rangle \rightarrow \langle B \rangle b \langle B \rangle$$

$$4. \langle A \rangle \rightarrow \langle D \rangle$$

$$10. \langle D \rangle \rightarrow \langle B \rangle$$

$$5. \langle B \rangle \rightarrow b \langle B \rangle$$

$$11. \langle E \rangle \rightarrow a b$$

$$6. \langle B \rangle \rightarrow d$$

Rešenje

a) Eliminacija mrtvih i nedostižnih neterminala

Prvo se eliminišu mrtvi neterminali. Određujemo skup živih neterminala \check{Z} . Inicijalno, skup \check{Z} uključuje one neterminale koji se pojavljuju na levim stranama smena na čijim desnim stranama se pojavljuju isključivo terminali:

$$\check{Z} = \{\langle B \rangle, \langle E \rangle\}$$

Zatim u skup \check{Z} dodajemo sve one neterminale koji se pojavljuju na levim stranama smena na čijim desnim stranama se pojavljuju samo terminali i oni neterminali koji se već nalaze u skupu \check{Z} . Ovaj postupak se iterativno ponavlja sve dok postoji promena skupa \check{Z} .

$$\check{Z} = \{\langle B \rangle, \langle E \rangle, \langle D \rangle\}$$

$$\check{Z} = \{ \langle B \rangle, \langle E \rangle, \langle D \rangle, \langle A \rangle \}$$

$$\check{Z} = \{ \langle B \rangle, \langle E \rangle, \langle D \rangle, \langle A \rangle \}$$

Po završetku formiranja skupa \check{Z} svi preostali neterminali koji se ne pojavljuju u \check{Z} pripadaju skupu mrtvih neterminala M .

$$M = \{ \langle C \rangle \}$$

Izbacujemo smene 2. i 7. u kojima se pojavljuje neterminal $\langle C \rangle$. Gramatika sada glasi:

$$1. \langle A \rangle \rightarrow \langle A \rangle c \langle B \rangle$$

$$8. \langle D \rangle \rightarrow \langle D \rangle a \langle B \rangle$$

$$3. \langle A \rangle \rightarrow c \langle D \rangle$$

$$9. \langle D \rangle \rightarrow \langle B \rangle b \langle B \rangle$$

$$4. \langle A \rangle \rightarrow \langle D \rangle$$

$$10. \langle D \rangle \rightarrow \langle B \rangle$$

$$5. \langle B \rangle \rightarrow b \langle B \rangle$$

$$11. \langle E \rangle \rightarrow a b$$

$$6. \langle B \rangle \rightarrow d$$

Zatim se eliminišu nedostižni neterminali. U tom cilju formirano skup dostižnih neterminala D koji se inicijalno sastoji od startnog neterminala.

$$D = \{ \langle A \rangle \}$$

Zatim u skup D dodajemo sve neterminale koji se pojavljuju na desnim stranama smena na čijim se levim stranama pojavljuju neterminali iz $\langle D \rangle$. Postupak se iterativno ponavlja sve dok postoji promena skupa D .

$$D = \{ \langle A \rangle, \langle B \rangle, \langle D \rangle \}$$

$$D = \{ \langle A \rangle, \langle B \rangle, \langle D \rangle \}$$

Kada je skup D formiran, preostali neterminali koji se ne pojavljuju u D pripadaju skupu nedostižnih neterminala ND .

$$ND = \{ \langle E \rangle \}$$

Iz gramatike izbacujemo smenu 11. u kojoj se pojavljuje neterminal $\langle E \rangle$. Gramatika glasi:

$$1. \langle A \rangle \rightarrow \langle A \rangle c \langle B \rangle$$

$$6. \langle B \rangle \rightarrow d$$

$$3. \langle A \rangle \rightarrow c \langle D \rangle$$

$$8. \langle D \rangle \rightarrow \langle D \rangle a \langle B \rangle$$

$$4. \langle A \rangle \rightarrow \langle D \rangle$$

$$9. \langle D \rangle \rightarrow \langle B \rangle b \langle B \rangle$$

$$5. \langle B \rangle \rightarrow b \langle B \rangle$$

$$10. \langle D \rangle \rightarrow \langle B \rangle$$

b)

Eliminacija leve rekurzije

Pravila transformacije smena:

$$\langle X \rangle \rightarrow \langle X \rangle \alpha_1$$

$$\langle X \rangle \rightarrow \beta_1 \langle X' \rangle$$

...

...

$$\langle X \rangle \rightarrow \langle X \rangle \alpha_m$$

$$\langle X \rangle \rightarrow \beta_n \langle X' \rangle$$

$$\langle X \rangle \rightarrow \beta_1 \quad \rightarrow$$

$$\langle X' \rangle \rightarrow \alpha_1 \langle X' \rangle$$

...

...

$$\langle X \rangle \rightarrow \beta_n$$

$$\langle X' \rangle \rightarrow \alpha_1 \langle X' \rangle$$

$$\langle X' \rangle \rightarrow \varepsilon$$

U datoj gramatici neterminali $\langle A \rangle$ i $\langle D \rangle$ su levo rekurzivni. Primenjujemo navedeno pravilo na smene za $\langle A \rangle$, pri čemu je $m = 1$ i $n = 2$, $\alpha_1 = c \langle B \rangle$, $\beta_1 = c \langle D \rangle$, $\beta_2 = \langle D \rangle$.

$$1. \langle A \rangle \rightarrow \langle A \rangle c \langle B \rangle$$

$$1'. \langle A \rangle \rightarrow c \langle D \rangle \langle A' \rangle$$

$$3. \langle A \rangle \rightarrow c \langle D \rangle \quad \rightarrow$$

$$2'. \langle A \rangle \rightarrow \langle D \rangle \langle A' \rangle$$

$$4. \langle A \rangle \rightarrow \langle D \rangle$$

$$3'. \langle A' \rangle \rightarrow c \langle B \rangle \langle A' \rangle$$

$$4'. \langle A' \rangle \rightarrow \varepsilon$$

Primenjujemo pravilo na smene za $\langle D \rangle$, pri čemu je $m = 1$ i $n = 2$, $\alpha_1 = a \langle B \rangle$, $\beta_1 = \langle B \rangle b \langle B \rangle$, $\beta_2 = \langle B \rangle$:

$$8. \langle D \rangle \rightarrow \langle D \rangle a \langle B \rangle$$

$$7'. \langle D \rangle \rightarrow \langle B \rangle b \langle B \rangle \langle D' \rangle$$

$$9. \langle D \rangle \rightarrow \langle B \rangle b \langle B \rangle \quad \rightarrow$$

$$8'. \langle D \rangle \rightarrow \langle B \rangle \langle D' \rangle$$

$$10. \langle D \rangle \rightarrow \langle B \rangle$$

$$9'. \langle D' \rangle \rightarrow a \langle B \rangle \langle D' \rangle$$

$$10'. \langle D' \rangle \rightarrow \varepsilon$$

Bitno je napomenuti da u ovoj gramatici nije bilo indirektnih rekurzija, jer $\langle B \rangle$ uvek počinje terminalom, a $\langle D \rangle$ nigde na svojoj desnoj strani nema pozivanje na $\langle A \rangle$, pa je zbog toga indirektna rekurzija nemoguća.

Gramatika sada ima oblik:

- | | |
|---|--|
| 1'. $\langle A \rangle \rightarrow c \langle D \rangle \langle A' \rangle$ | 6'. $\langle B \rangle \rightarrow d$ |
| 2'. $\langle A \rangle \rightarrow \langle D \rangle \langle A' \rangle$ | 7'. $\langle D \rangle \rightarrow \langle B \rangle b \langle B \rangle \langle D' \rangle$ |
| 3'. $\langle A' \rangle \rightarrow c \langle B \rangle \langle A' \rangle$ | 8'. $\langle D \rangle \rightarrow \langle B \rangle \langle D' \rangle$ |
| 4'. $\langle A' \rangle \rightarrow \epsilon$ | 9'. $\langle D' \rangle \rightarrow a \langle B \rangle \langle D' \rangle$ |
| 5'. $\langle B \rangle \rightarrow b \langle B \rangle$ | 10'. $\langle D' \rangle \rightarrow \epsilon$ |

c)

Leva faktORIZACIJA

Pravilo leve faktORIZACIJE glasi:

$$\begin{array}{lcl} \langle S \rangle \rightarrow \alpha \beta & & \langle S \rangle \rightarrow \alpha \langle S' \rangle \\ \langle S \rangle \rightarrow \alpha \gamma & \rightarrow & \langle S' \rangle \rightarrow \beta \\ & & \langle S' \rangle \rightarrow \gamma \end{array}$$

Primenom ovog pravila na smene 7'. i 8'., pri čemu je $\alpha =$

$$\begin{array}{lcl} 7'. \langle D \rangle \rightarrow \langle B \rangle b \langle B \rangle \langle D' \rangle & \rightarrow & 7''. \langle D \rangle \rightarrow \langle B \rangle \langle D'' \rangle \\ 8'. \langle D \rangle \rightarrow \langle B \rangle \langle D' \rangle & & 8''. \langle D'' \rangle \rightarrow b \langle B \rangle \langle D' \rangle \\ & & 9''. \langle D'' \rangle \rightarrow \langle D' \rangle \end{array}$$

Na kraju će gramatika imati sledeći oblik:

- | | |
|--|---|
| 1''. $\langle A \rangle \rightarrow c \langle D \rangle \langle A' \rangle$ | 7''. $\langle D \rangle \rightarrow \langle B \rangle \langle D'' \rangle$ |
| 2''. $\langle A \rangle \rightarrow \langle D \rangle \langle A' \rangle$ | 8''. $\langle D'' \rangle \rightarrow b \langle B \rangle \langle D' \rangle$ |
| 3''. $\langle A' \rangle \rightarrow c \langle B \rangle \langle A' \rangle$ | 9''. $\langle D'' \rangle \rightarrow \langle D' \rangle$ |
| 4''. $\langle A' \rangle \rightarrow \epsilon$ | 10''. $\langle D' \rangle \rightarrow a \langle B \rangle \langle D' \rangle$ |
| 5''. $\langle B \rangle \rightarrow b \langle B \rangle$ | 11''. $\langle D' \rangle \rightarrow \epsilon$ |
| 6''. $\langle B \rangle \rightarrow d$ | |

d) Određivanje selekcionih skupova

$$\begin{aligned} \text{SELECT}(1'') &= \{c\} \\ \text{SELECT}(2'') &= \text{FIRST}(\langle D \rangle) = \text{FIRST}(\langle B \rangle) = \{b, d\} \\ \text{SELECT}(3'') &= \{c\} \\ \text{SELECT}(4'') &= \text{FOLLOW}(\langle A' \rangle) = \text{FOLLOW}(\langle A \rangle) = \{\epsilon\} \\ \text{SELECT}(5'') &= \{b\} \\ \text{SELECT}(6'') &= \{d\} \\ \text{SELECT}(7'') &= \text{FIRST}(\langle B \rangle) = \{b, d\} \\ \text{SELECT}(8'') &= \{b\} \\ \text{SELECT}(9'') &= \text{FIRST}(\langle D' \rangle) \cup \text{FOLLOW}(\langle D'' \rangle) = \{a\} \cup \text{FOLLOW}(\langle D \rangle) \\ &= \{a\} \cup \text{FIRST}(\langle A' \rangle) \cup \text{FOLLOW}(\langle A \rangle) \\ &= \{a\} \cup \{c\} \cup \{\epsilon\} \\ &= \{a, c, \epsilon\} \\ \text{SELECT}(10'') &= \{a\} \\ \text{SELECT}(11'') &= \text{FOLLOW}(\langle D' \rangle) = \text{FOLLOW}(\langle D'' \rangle) = \text{FOLLOW}(\langle D \rangle) \\ &= \text{FIRST}(\langle A' \rangle) \cup \text{FOLLOW}(\langle A \rangle) \\ &= \{c, \epsilon\} \end{aligned}$$

Ova gramatika jeste LL (1) tipa jer su selekcionni skupovi disjunktni.

Zadatak 4.

Transforisati sledeće gramatike u LL(1) gramatike:

- | | | |
|--|--|--|
| 1. $\langle S \rangle \rightarrow a b \langle A \rangle b$ | 1. $\langle S \rangle \rightarrow b c e$ | 1. $\langle S \rangle \rightarrow \langle S \rangle xyz$ |
| 2. $\langle S \rangle \rightarrow a c \langle S \rangle a$ | 2. $\langle S \rangle \rightarrow \langle B \rangle c \langle S \rangle a$ | 2. $\langle S \rangle \rightarrow ab$ |
| | 3. $\langle B \rangle \rightarrow b d$ | 3. $\langle S \rangle \rightarrow cd$ |
| | 4. $\langle B \rangle \rightarrow d$ | |

Rešenje:

a) Pošto smene 1 i 2 imaju iste selekzione skupove i počinju terminalima, korišćenjem leve faktorizacije gramatika se svodi na LL(1) gramatiku.

1. $\langle S \rangle \rightarrow a \langle X \rangle$
2. $\langle X \rangle \rightarrow b \langle A \rangle b$
3. $\langle X \rangle \rightarrow c \langle S \rangle a$

b) Smene 1 i 2 imaju iste selekzione skupove. Pošto smena 2 počinje neterminalom $\langle B \rangle$ umesto njega se ubacuju sve desne strane smena u kojima je $\langle B \rangle$ na levoj strani:

1. $\langle S \rangle \rightarrow b c$
2. $\langle S \rangle \rightarrow b d c \langle S \rangle a$
3. $\langle S \rangle \rightarrow d c \langle S \rangle a$

Levom faktorizacijom se eliminišu isti selekциони skupovi u smenama 1 i 2.

c) Eliminisanje leve rekurzije:

1. $\langle S \rangle \rightarrow ab \langle S' \rangle$
2. $\langle S \rangle \rightarrow cd \langle S' \rangle$
3. $\langle S' \rangle \rightarrow xyz \langle S' \rangle$

Zadatak 5

Napisati gramatiku koja je

- a) LL(1) ali nije LR(0)
- b) LR(0) ali nije LL(1)
- c) LL(1) ali nije SLR(1).

Rešenje

a) Rešenje predstavlja bilo koja LL(1) gramatika sa praznom smenom, s obzirom da LR(0) gramatike ne poseduju prazne smene, na primer:

1. $\langle S \rangle \rightarrow a$ $\text{SELECT}(1) = \{a\}$
2. $\langle S \rangle \rightarrow \varepsilon$ $\text{SELECT}(2) = \{\mid\}$

b) Rešenje predstavlja bilo koja LR(0) gramatika sa levom rekurzijom, na primer LR(0) – proveriti konstrukcijom parsera)

1. $\langle S \rangle \rightarrow \langle S \rangle a$ $\text{SELECT}(1) = \{a\}$
2. $\langle S \rangle \rightarrow a$ $\text{SELECT}(2) = \{a\}$

c) Svaka LL(1) gramatika nije nužno i SLR(1) gramatika. Dok se kod LR(1) gramatika pri svođenju ručke na neki neterminal $\langle A \rangle$ gleda tačan kontekst u kome se neterminal $\langle A \rangle$ javlja, kod SLR(1)

gramatika stvar se uprošćava korišćenjem skupa FOLLOW(<A>). Sledi primer LL(1) gramatike u kome se <A> javlja dva puta u različitom kontekstu – prvo dešavanje je praćeno terminalom a, a drugo terminalom b.

1. $\langle S \rangle \rightarrow \langle A \rangle a \langle A \rangle b$ $\text{SELECT}(1) = \text{FIRST}(\langle A \rangle) \cup \{a\} = \emptyset \cup \{a\} = \{a\}$
2. $\langle S \rangle \rightarrow b$ $\text{SELECT}(2) = \{b\}$
3. $\langle A \rangle \rightarrow \varepsilon$ $\text{SELECT}(3) = \text{FOLLOW}(\langle A \rangle) = \{a, b\}$

Kod SLR(1) parsera, u vrsti ∇ i koloni b kontrolne tabele javlja se SHIFT/REDUCE(3) konflikt:

∇	
$\langle S' \rangle \rightarrow \bullet \langle S \rangle - $	
$\langle S \rangle \rightarrow \bullet \langle A \rangle a \langle A \rangle b$	
$\langle S \rangle \rightarrow \bullet b$	
$\langle A \rangle \rightarrow \varepsilon \bullet$	

Zadatak 6. (Primer sa predavanja)

Konstruisati parser na principu rekurzivnog spusta koji odgovara prepoznavaću aritmetičkog izraza pomoću sledećih produkcija.

1. $\langle E \rangle \rightarrow \langle T \rangle \langle E_list \rangle$
2. $\langle E_list \rangle \rightarrow + \langle T \rangle \langle E_list \rangle$
3. $\langle E_list \rangle \rightarrow \varepsilon$
4. $\langle T \rangle \rightarrow \langle P \rangle \langle T_list \rangle$
5. $\langle T_list \rangle \rightarrow * \langle P \rangle \langle T_list \rangle$
6. $\langle T_list \rangle \rightarrow \varepsilon$
7. $\langle P \rangle \rightarrow (\langle E \rangle)$
8. $\langle P \rangle \rightarrow I$

Analiza problema

Realizaciju parsera u višem programskom jeziku možemo sprovesti tako da umesto potisnog automata koji isključivo radi sa sadržajem steka i ulaza, napišemo program koga čini skup uzajamno rekurzivnih potprograma. U tom slučaju ulogu steka potisnog automata preuzima izvršni stek programa. Ovo rešenje pogodno je u slučaju gramatika sa manjim brojem smena i neterminala.

Rešenje

Programsko rešenje čini glavni program i niz uzajamno rekurzivnih potprograma. Za svaki neterminal gramatike obezbeđuje se jedan potprogram (u konkretnom slučaju imaćemo potprograme PROCE, PROCE_LIST, PROCT, PROCT_LIST i PROCP).

Potprogram za određeni neterminal ima ulogu da prepozna primenu neke od smena sa datim neterminalom na levoj strani i upari neterminal sa ulazom u skladu sa prepoznatom smenom. U priloženom pseudoprogramu globalna promenljiva IN predstavlja tekući ulazni znak. U potprogramu PROCP, korišćenjem CASE konstrukcije, prepoznaje se, na primer, smena 7 u slučaju da je tekući ulazni znak (. Ovo je naravno uslovljeno selekcionim skupom 7. smene. Sledeći fragment koda odgovara tom slučaju, odnosno uparivanju desne strane smene $\langle P \rangle \rightarrow (\langle E \rangle)$ sa tekućim ulazom:

```
P7:   IN = NEXTCHAR();
      call PROCE;
      if ( IN = '(' )
          then IN = NEXTCHAR();
          return;
      else REJECT;
      end if
```

U ovom fragmentu, sa ulaza se čita sledeći znak pozivom funkcije NEXTCHAR() i smešta u IN (time je uparena i sa ulaza sklonjena otvorena zagrada), zatim se pozivom PROCE ostvaruje uparivanje neterminala <E> sa delom preostalog ulaza i potom se ispituje da li je posle ovoga uparivanja tekući znak ')', ukoliko jeste ovaj znak se sklanja sa ulaza. Time je cela desna strana 7. smene uparena sa ulazom.

Glavni program inicijalizuje promenljivu IN da sadrži prvi ulazni znak. Zatim, pozivom potprograma PROCE, inicira uparivanje ulazne sekvence sa startni neterminalom. Po povratku iz potprograma PROCE u glavnom programu ispituje se da li je procesiran ceo ulaz što označava da se ulazna sekvenca prihvata. U suprotnom, ako tekući znak nije marker kraja, sekvenca se odbija.

Glavni program: IN = NEXTCHAR(); call PROCE; if (IN = '—') then ACCEPT; else REJECT; end if; end; procedure PROCE: case IN of 'I', '(': goto P1; '+', '*', ')', '—': REJECT; end case; P1: call PROCT; call PROCE_LIST; return; end procedure; procedure PROCP: case IN of 'I' : goto P8; '(' : goto P7; '+', '*', ')', '—': REJECT; end case; P8: IN = NEXTCHAR(); return; P7: IN = NEXTCHAR(); call PROCE; if (IN = ')') then IN = NEXTCHAR(); return; else REJECT; end if; end procedure;	procedure PROCT: case IN of 'I', '(': goto P4; '+', '*', ')', '—': REJECT; end case; P4 : call PROCP; call PROCT_LIST; return; end procedure; procedure PROCE_LIST: case IN of '+': goto P2; ')', '—': goto P3; 'I', '(', '*' : REJECT; end case; P2: IN = NEXTCHAR(); call PROCT; call PROCE_LIST; return; P3: return; end procedure; procedure PROCT_LIST: case IN of '*' : goto P5; '+', ')', '—': goto P6; 'I', '(' : REJECT; end case; P5: IN = NEXTCHAR(); call PROCP; call PROCT_LIST; return; P6: return; end procedure;
---	--

Zadatak 7.

Za datu gramatiku konstruisati LL(1) parser rekurzivnim spustom.

1. <S> → b <A>	3. <A> → c <S>
2. <S> → a <S>	4. <A> → d <A> a

Analiza problema:

Pretpostavimo da postoje pomoćne funkcije:

char current() - koja vraća trenutni karakter na ulazu

void advance() - koja prelazi na sledeći karakter na ulazu

```
void terminal(char t)
{
    if(current() == t) advance();
    else error;
}
```

Rešenje:

Za svaki neterminal, potrebno je napisati jednu proceduru koja u zavisnosti od trenutnog ulaza i selekcionih skupova smena, odlučuje koja smena će biti pozvana.

```
void S() {
    switch(current()) {
        case 'b' : P1();
        case 'a' : P2();
        default : error;
    }
}

void A() {
    switch(current()) {
        case 'c' : P1();
        case 'd' : P2();
        default : error;
    }
}
```

Za svaku smenu, potrebno je napisati jednu proceduru koja ispituje da li je redosled gramatičkih simbola ispravan.

<pre>void P1() { terminal('b'); A(); }</pre>	<pre>void P3() { terminal('c'); A(); }</pre>
<pre>void P2() { terminal('a'); S(); }</pre>	<pre>void P4() { terminal('b'); A(); terminal('c'); }</pre>

Pozivom procedure startnog neterminala ulaz se može parsirati u skladu sa gramatikom. Posle povratka iz procedure S() tekući ulaz treba da je EOF.

Zadatak 8.

a) Kojoj gramatici odgovara sledeći rekurzivni prepoznavać napisan na pseudojeziku:

<pre>glavni program: INP = prvi ulazni simbol call PROCS; if (INP < '⊥') then REJECT; else ACCEPT; end if; end; procedure PROCS: case INP of 'a', 'c', 'd': {call PROCA; if (INP < 'a') then REJECT else call ADVANCE; call PROCA; end if; } 'b': call ADVANCE; '⊥': REJECT; end case; return; end procedure;</pre>	<pre>procedure PROCA: case INP of 'c' : { call PROCA; if (INP < 'c') then REJECT else ADVANCE; end if; } 'd' : call ADVANCE; 'a', '⊥': /* ništa */; 'b': REJECT; end case; return; end procedure; procedure ADVANCE: INP = sledeći ulazni simbol; return; end procedure;</pre>
--	--

b) Nalaženjem SELECT skupova odgovoriti na pitanje da li je gramatika iz tačke a) tipa LL(1).

Rešenje

U rekurzivnom prepoznavaću svakom neterminalu u gramatici odgovara jedna procedura. U konkretnom slučaju gramatika ima dva neterminala $\langle S \rangle$ i $\langle A \rangle$ pri čemu je $\langle S \rangle$ startni neterminal s obzirom da se odgovarajuća procedura PROCS poziva iz glavnog programa. Analizom procedure PROCS nalazimo smene koje imaju $\langle S \rangle$ na levoj strani:

- Programski fragment:

```
case INP of
  'a', 'c', 'd': {call PROCA;
                  if (INP < 'a')
                    then REJECT
                    else call ADVANCE;
                  call PROCA;
                  end if;
                }
odgovara smeni  $\langle S \rangle \rightarrow \langle A \rangle$  a  $\langle A \rangle$ 
```

- Programski fragment:

```
'b': call ADVANCE;
odgovara smeni  $\langle S \rangle \rightarrow b$ 
```

Ovo su jedine smene koje odgovaraju neterminalu $\langle S \rangle$. Analizom PROCA utvrđujemo smene za $\langle A \rangle$:

- Programski fragment:

```

case INP of
  'c' : { call PROCA;
        if (INP <> 'c')
          then REJECT
          else ADVANCE;
        end if;
      }
odgovara smeni <A> → <A> c

```

- Programski fragment:

```

  'd' : call ADVANCE;
odgovara smeni <A> → d

```

- Programski fragment

```

  'a', '⊥' : /* ništa */;
odgovara smeni <A> → ε

```

Prema tome, gramatika na osnovu koje je napravljen program ima sledeći izgled:

1. $\langle S \rangle \rightarrow \langle A \rangle a \langle A \rangle$
2. $\langle S \rangle \rightarrow b$
3. $\langle A \rangle \rightarrow \langle A \rangle c$
4. $\langle A \rangle \rightarrow d$
5. $\langle A \rangle \rightarrow \varepsilon$

b)

Neterminal $\langle A \rangle$ je poništiv, a neterminal $\langle S \rangle$ nije, tako da imamo:

```

FIRST(⟨A⟩) = FIRST(⟨A⟩) ∪ {c} ∪ {d} = {c, d}
FIRST(⟨S⟩) = FIRST(⟨A⟩) ∪ {a} ∪ {b} = {a, b, c, d}
FOLLOW(⟨S⟩) = {⊥}
FOLLOW(⟨A⟩) = {a} ∪ FOLLOW(⟨S⟩) ∪ {c} = {a, c, ⊥}

```

Sada se mogu odrediti selekcionni skupovi smena:

```

SELECT(1) = FIRST(⟨A⟩ a ⟨A⟩) = FIRST(⟨A⟩) ∪ {a} = {a, c, d}
SELECT(2) = {b}
SELECT(3) = FIRST(⟨A⟩ c) = FIRST(⟨A⟩) ∪ {c} = {c, d}
SELECT(4) = {d}
SELECT(5) = FOLLOW(⟨A⟩) = {a, c, ⊥}

```

Vidimo da selekcionni skupovi 3., 4. i 5. smene nisu disjunktne (što se moglo očekivati s obzirom da smena 3. poseduje direktnu levu rekurziju), pa gramatika nije LL(1). Samim tim, ni zadati rekurzivni prepoznavać ne radi ispravno, odnosno, može da uđe u beskonačnu rekurziju ako je c tekući ulazni simbol u trenutku izvršavanja PROCA.